INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Photometric visual servoing

Christophe Collewet — Eric Marchand

N° 6631

Septembre 2008

Thème COG

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6631--FR+ENG

Photometric visual servoing

Christophe Collewet*, Eric Marchand†

Thème COG — Systèmes cognitifs
Équipe-Projet Lagadic

Rapport de recherche n° 6631 — Septembre 2008 — 36 pages

Abstract: This report proposes a new way to achieve robotic tasks by 2D visual servoing. Indeed, instead of using classical geometric features such as points, straight lines, pose or an homography, as it is usually done, the luminance of all pixels in the image is here considered. The main advantage of this new approach is that it does not require any tracking or matching process. The key point of our approach relies on the analytic computation of the interaction matrix that links the time variation of the luminance to the camera motions. This computation is based either on a simple Lambertian model or on the Phong one so that complex illumination changes can be considered. However, since most of the classical control laws fail when considering the luminance as a visual feature, we turn the visual servoing problem into an optimization one leading to a new control law. Experimental results on positioning and tracking tasks validate the proposed approach and show its robustness regarding to approximated depths, Lambertian and non Lambertian objects, low textured objects and partial occlusions.

Key-words: Visual servoing, photometry, illumination model, tracking, optimization

Part of this report has been published in the IEEE Int. Conf. on Robotics and Automation, ICRA'08 and IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'08.

* Christophe.Collewet@irisa.fr
† Eric.Marchand@irisa.fr

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

Asservissement visuel photométrique

Résumé : Nous décrivons dans ce rapport une nouvelle façon de réaliser des tâches robotiques par asservissement visuel 2D. En effet, au lieu d'utiliser des informations visuelles de nature géométrique, comme par exemple des points, des lignes droites, la pose ou une homographie, comme c'est habituellement le cas, la luminance en chaque pixel de l'image est considérée. L'avantage principal de cette nouvelle approche réside dans le fait qu'aucune phase de suivi ou de mise en correspondance n'est requise. Le point clé de cette approche repose sur l'obtention sous forme analytique de la matrice dite d'interaction, matrice liant la variation temporelle de la luminance au mouvement de la caméra. Ce calcul est basé sur le simple modèle d'illumination de Lambert ou sur le modèle de Phong afin que des variations complexes d'illumination puissent être appréhendées. Cependant, les lois de commande habituellement utilisées étant dans ce cas mises en échec, nous reformulons le problème de l'asservissement visuel comme un problème d'optimisation aboutissant à l'écriture d'une nouvelle loi de commande. Des résultats expérimentaux, aussi bien concernant la réalisation de tâches de positionnement que de suivis de cible, valident l'approche proposée et montrent sa robustesse vis-à-vis de l'approximation faite sur les profondeurs, de scènes non Lambertiennes, de scènes peu texturées ou encore partiellement occultées.

Mots-clés : Asservissement visuel, luminance, modèle d'illumination, suivi, optimisation

Sommaire

1 Introduction

Visual servoing consists in using information provided by a vision sensor for the control of a robot [1]. Robust extraction and real-time spatio-temporal tracking of visual cues is then usually one of the keys to success of a visual servoing task. In this report we show that this tracking process can be totally removed and show that no other information than the image intensity (the pure luminance signal) can be considered to control the robot motion.

Classically, to achieve a visual servoing task, a set of visual features has to be selected from the image allowing to control the desired degrees of freedom. A control law has also to be designed so that these visual features $\mathbf{s}$ reach a desired value $\mathbf{s}^*$, leading to a correct realization of the task. The control principle is thus to regulate to zero the error vector $\mathbf{s} - \mathbf{s}^*$. To build this control law, the knowledge of the interaction matrix $\mathbf{L_s}$ is usually required. For eye-in-hand systems, this matrix links the time variation of $\mathbf{s}$ to the camera instantaneous velocity $\mathbf{v}$

$$\dot{\mathbf{s}} = \mathbf{L_s}\,\mathbf{v} \tag{1}$$

with $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$ where $\boldsymbol{v}$ is the linear camera velocity and $\boldsymbol{\omega}$ its angular velocity. Thereafter, if we consider the camera velocity as input of the robot controller, the following control law is designed to try to obtain an exponential decoupled decrease of the error $\mathbf{s} - \mathbf{s}^*$

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}_{\mathbf{s}}^{+} \left(\mathbf{s} - \mathbf{s}^*\right) \tag{2}$$

where λ is a proportional gain that has to be tuned to minimize the time-to-convergence, and $\widehat{\mathbf{L}}_{\mathbf{s}}^{+}$ is the pseudo-inverse of a model or an approximation of $\mathbf{L_s}$ [1].

As can be seen, visual servoing explicitly relies on the choice of the visual features $\mathbf{s}$ (and then on the related interaction matrix); that is the key point of this approach. However, with a vision sensor providing 2D measurements $\mathbf{x}(\mathbf{r}_k)$ (where $\mathbf{r}_k$ is the camera pose at time k), potential visual features $\mathbf{s}$ are numerous, since 2D data (coordinates of feature points in the image, contours, moments,...) as well as 3D data provided by a localization algorithm exploiting $\mathbf{x}(\mathbf{r}_k)$ can be considered. In all cases, if the choice of $\mathbf{s}$ is important, it is always designed from the visual measurements $\mathbf{x}(\mathbf{r}_k)$. However, a robust extraction, matching (between $\mathbf{x}(\mathbf{r}_0)$ and $\mathbf{x}^* = \mathbf{x}(\mathbf{r}^*)$) and real-time spatio-temporal tracking (between $\mathbf{x}(\mathbf{r}_{k-1})$ and $\mathbf{x}(\mathbf{r}_k)$) have proved to be a complex task, as testified by the abundant literature on the subject (see [2] for a recent survey on this subject). This image processing is, to date, a necessary step and considered also as one of the bottlenecks of the expansion of visual servoing. That is why some works tend to alleviate this problem. A first idea is to select visual features as proposed in [3, 4] or as in [5] to only keep visual features that are tracked with a high confident level (see also [6] where a more general approach is proposed). However, the goal of such approaches is not to simplify the image processing step but to take into account that it can fail. A more interesting way to avoid any tracking process is to use non geometric visual features. In that case, parameters of a 2D motion model are used as in [7–10]. Nevertheless, such approaches require an important and complex image processing step. Removing the entire matching process is only possible when using directly the luminance as we propose.

Indeed, to achieve this goal we use as visual features the simplest feature that can be considered: the image intensity itself. We therefore called this new approach *photometric visual servoing*. In that case, the visual feature vector $\mathbf{s}$ is nothing but the image while $\mathbf{s}^*$ is the desired image. The error $\mathbf{s} - \mathbf{s}^*$ is then only the difference between the current and desired images (that is $\mathbf{I} - \mathbf{I}^*$ where $\mathbf{I}$ is a vector that contains image intensity of all pixels). Within this framework, the major contributions of this report are:

- The analytic computation of the interaction matrix $\mathbf{L_I}$ related to the luminance as well in the case of the temporal luminance constancy hypothesis as in the case of complex illumination changes.
- The approach requires no matching, no tracking and very few image processing process.
- Using the image intensity as visual features, the classical control law, given by equation (2), at best converges with a slow and inappropriate camera motion or simply diverges, we thus turn the visual servoing problem into a minimization one.
- Positioning and tracking tasks that control the 6 d.o.f of the camera are considered.

Considering the whole image as a feature has previously been considered [11, 12]. As in our case, the methods presented in [11, 12] did not require a matching process. Nevertheless they differ from our approach in two important points. First, they do not use directly the image intensity but an eigenspace decomposition is performed to reduce the dimensionality of image data. The control is then performed in the eigenspace and not directly with the image intensity. Moreover, this way to proceed requires the off-line computation of this eigenspace and then, for each new frame, the projection of the image on this subspace. Second, the interaction matrix related to the eigenspace is not computed analytically but learned during an off-line step. This learning process has two drawbacks: it has to be done for each new object and requires the acquisition of many images of the scene at various camera positions. Considering an analytical interaction matrix avoids these issues.

An interesting approach, which also consider the pixels intensity, has been recently proposed in [13]. This approach is based on the use of kernel methods that lead to a high decoupled control law. However, only the translations and the rotation around the optical axis are considered whereas, in our work, the 6 degrees of freedom are controlled. Another approach that does not require tracking nor matching has been proposed in [14]. It models collectively feature points extracted from the image as a mixture of Gaussian and try to minimize the distance function between the Gaussian mixture at current and desired positions. Simulation results show that this approach is able to control the 3 d.o.f. of robot (and the 6 d.o.f. under some assumptions). However, note that an image processing step is still required to extract the current feature points. Our approach does not require this step. Finally, in [15], the authors present an homography-based approach to visual servoing. In this method the image intensity of a planar patch is first used to estimate the homography (using the ESM algorithm described in [15] for example) between current and desired image which is then used to build the control law. Despite the fact that, as in our case, image intensity is used as the basis of the approach, an important image processing step is necessary to estimate the homography. Furthermore, the visual features used in the control law rely on the homography matrix and not directly on the luminance.

In the remainder of this report we first compute the interaction matrix related to the luminance in the general case in Section 2 and in some particular cases in Section 3. Then, we reformulate the visual servoing problem into an optimization problem in

Section 4 and propose a new control law dedicated to the specific case of the luminance. Section 6 shows experimental results on various scenes for several tasks.

2 Luminance as a visual feature

The visual features considered in this report are the luminance I of each point of the image. We have

$$\mathbf{s}(\mathbf{r}) = \mathbf{I}(\mathbf{r}) = (\mathbf{I}_{1\bullet}, \mathbf{I}_{2\bullet}, \cdots, \mathbf{I}_{N\bullet}) \tag{3}$$

where $\mathbf{I}_{k\bullet}$ is nothing but the k-th line of the image. $\mathbf{I}(\mathbf{r})$ is then a vector of size $N \times M$ where $N \times M$ is the size of the image. An estimation of the interaction matrix is at the center of the development of any visual servoing process. In our case, we are looking for the interaction matrix related to the luminance of a pixel in the image, that is

$$\lim_{dt \to 0} \frac{I(\mathbf{x}, t + dt) - I(\mathbf{x}, t)}{dt} = \mathbf{L}_I(\mathbf{x})\mathbf{v} \tag{4}$$

$\mathbf{x} = (x, y)$ being the normalized coordinates of the projection of a point $\mathbf{P}$ belonging to the scene.

2.1 Interaction matrix under temporal luminance constancy

Before computing the interaction matrix $\mathbf{L}_I(\mathbf{x})$ in the general case, we first consider the simpler case where the temporal luminance constancy hypothesis is assumed as it is done in most of computer vision applications:

$$I(\mathbf{x} + \mathbf{dx}, t + dt) = I(\mathbf{x}, t) \tag{5}$$

where $\mathbf{x}$ denotes the normalized coordinates of the perspective projection $\mathbf{p}$ of a physical point $\mathbf{P}$ and assuming that $\mathbf{p}$ has a small displacement $\mathbf{dx}$ in the time interval dt. If $\mathbf{dx}$ is small enough, a first order Taylor series expansion of (5) around $\mathbf{x}$ can be performed yielding the so-called *optical flow constraint equation* (OFCE) [16]

$$\boldsymbol{\nabla} I^\top \dot{\mathbf{x}} + I_t = 0 \tag{6}$$

with $\boldsymbol{\nabla} I$ the spatial gradient of $I(\mathbf{x}, t)$[1] and $I_t = \partial I(\mathbf{x}, t)/\partial t$. Moreover, considering the interaction matrix $\mathbf{L}_\mathbf{x}$ related to $\mathbf{x}$ (i.e. $\dot{\mathbf{x}} = \mathbf{L}_\mathbf{x}\mathbf{x}$)

$$\mathbf{L}_\mathbf{x} = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \tag{7}$$

relation (6) gives

$$I_t = -\nabla I^\top \mathbf{L}_\mathbf{x}\mathbf{v}. \tag{8}$$

However, note that I_t is nothing but the left part of (4). Consequently, from (4) and (8), we obtain the interaction matrix $\mathbf{L}_I(\mathbf{x})$ related to I at pixel $\mathbf{x}$

$$\mathbf{L}_I(\mathbf{x}) = -\nabla I^\top \mathbf{L}_\mathbf{x}. \tag{9}$$

Of course, because of the hypothesis required to derive (5), (9) can only be valid for Lambertian scenes, that is for surfaces reflecting the light with the same intensity in each direction. Besides, (9) is also only valid for a motionless lighting source with respect to the scene.

[1] Let us point out that the computation of $\boldsymbol{\nabla} I$ is the *only* image processing step necessary to implement our method.

2.2 Interaction matrix in the general case

In fact, it is well known that the constraint (5) can be easily violated [17], e.g. if the orientation of a Lambertian surface is changing with respect to the lighting. Consequently, many authors have addressed this problem in the context of computer vision. However, most of these works requires a off-line learning step and relie on the fact that an image of a Lambertian surface, without selfshadowing, can be linearly expressed from 3 images acquired at the same location under various illumination [18, 19]. This interesting property is used for example in [20–22]. This approach is extended in [23] to non-Lambertian scenes where the intensity variation between dt is expressed as a mixture of causes, however without relying on a physical description of the reflection phenomenon. This latter work can be seen as related to [24] since the illumination changes are modeled as a surface that evolves over time. A general framework has been proposed in [25] leading to the following relation:

$$I(\mathbf{x} + \mathbf{dx}, t + dt) = \big(1 + m(\mathbf{x}, t)\big) I(\mathbf{x}, t) + c(\mathbf{x}, t). \tag{10}$$

However, the coefficients $m(\mathbf{x}, t)$ and $c(\mathbf{x}, t)$ are considered as locally constant, and then are estimated numerically. Nevertheless, all these works do not rely on a time-varying physical reflection model.

A very interesting approach, close to our work, can be found in [26]. Indeed, a generalization of (6) is presented leading to

$$\boldsymbol{\nabla} I^\top \dot{\mathbf{x}} + I_t = \frac{dI}{dt} \tag{11}$$

where dI/dt is analytically computed according to several physical models of luminance variation. In particular, they study the case of a non planar Lambertian surface undergone to a rotation motion.

As already stated, to derive the interaction matrix, we have to consider a more realistic reflection model than the Lambert's one. Indeed, the Lambert's model can only explained the behavior of non homogeneous opaque dielectric material [27]. It only describes a diffuse reflection component and does not take into account the viewing direction. The Beckmann-Spizzichino model [28] is based on the electromagnetism laws and on the modeling of the surface asperities. It can thus take into account electrical conductors or not, either if the surface is smooth or rough. Concerning the Torrance-Sparrow model [29], it is based on the geometrical optics equations, and thus it is restricted to cases where the size of the surface asperities widely exceeds the wavelength of the incident light. This model can thus not be used with rough material. However, all these models are based on numerous unknowns parameters that are difficult to extract on line. That is why prefer to use a simpler model, the Phong one [30]; contrary to the previous models, this model is not based on physical laws, but comes from the computer graphics community. Although empirical, this model is widely used thanks to its simplicity, and because it is appropriate for various types of materials, whether they are rough or smooth. Note that other models could be considered such as the Blinn-Phong [31] as reported in [32].

According to the Phong model (see Fig. 1), the intensity $I(\mathbf{x})$ at point $\mathbf{x}$ writes as follows

$$I(\mathbf{x}) = K_s \cos^k \alpha + K_d \cos \theta + K_a. \tag{12}$$

This relation is composed of a diffuse, a specular and an ambient component and assumes a point light source. The scalar K_s describes the specular component of the

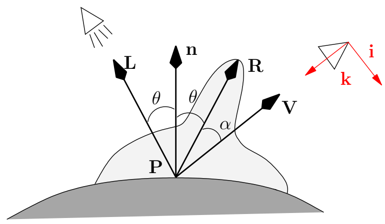


Figure 1: The Phong illumination model [30].

lighting; K_d describes the part of the diffuse term which depends on the *albedo* in $\mathbf{P}$; K_a is the intensity of ambient lighting in $\mathbf{P}$. Note that K_s, K_d and K_a depend on $\mathbf{P}$. θ is the angle between the normal to the surface $\mathbf{n}$ in $\mathbf{P}$ and the direction of the light source $\mathbf{L}$; α is the angle between $\mathbf{R}$ (which is $\mathbf{L}$ mirrored about $\mathbf{n}$) and the viewing direction $\mathbf{V}$. $\mathbf{R}$ can be seen as the direction due to a pure specular object, where k allows to model the width of the specular lobe around $\mathbf{R}$, this scalar varies as the inverse of the roughness of the material.

In the remainder of this report, the unit vectors $\mathbf{i}, \mathbf{j}$ and $\mathbf{k}$ correspond to the axis of the camera frame (see Fig. 1).

Considering that $\mathbf{R}$, $\mathbf{V}$ and $\mathbf{L}$ are normalized, we can rewrite (12) as

$$I(\mathbf{x}) = K_s {u_1}^k + K_d u_2 + K_a \tag{13}$$

where $u_1 = \mathbf{R}^\top \mathbf{V}$ while we have $u_2 = \mathbf{n}^\top \mathbf{L}$. Note that these vectors are easy to compute, since we have

$$\begin{aligned} \mathbf{V} &= -\frac{\mathbf{x}}{\| \mathbf{x} \|} && (14) \\ \mathbf{R} &= 2u_2\mathbf{n} - \mathbf{L}. && (15) \end{aligned}$$

In the general case, we consider the following dependences

$$\left\{ \begin{array}{lcl} \mathbf{V} &=& \mathbf{V}\big(\mathbf{x}(t)\big) \\ \mathbf{n} &=& \mathbf{n}\big(\mathbf{x}(t), t\big) \\ \mathbf{L} &=& \mathbf{L}\big(\mathbf{x}(t), t\big) \\ \mathbf{R} &=& \mathbf{R}\big(\mathbf{x}(t), t\big). \end{array} \right. \tag{16}$$

From the definition of the interaction matrix given in (4), its computation requires to write the total derivative of (13)

$$\dot{I} = kK_s u_1^{k-1} \dot{u}_1 + K_d \dot{u}_2. \tag{17}$$

However, it is also possible to compute $\dot{I}$ as

$$\dot{I} = \boldsymbol{\nabla} I^\top \dot{\mathbf{x}} + I_t = \boldsymbol{\nabla} I^\top \mathbf{L}_{\mathbf{x}} \mathbf{v} + I_t \tag{18}$$

where we have introduced the interaction matrix $\mathbf{L}_{\mathbf{x}}$ associated to $\mathbf{x}$. Consequently, from (17) and (18), we obtain

$$I_t + \boldsymbol{\nabla} I^\top \mathbf{L}_{\mathbf{x}} \mathbf{v} = kK_s u_1^{k-1} \dot{u}_1 + K_d \dot{u}_2 \tag{19}$$

that is a general formulation of the OFCE considering the Phong illumination model.

Thereafter, by explicitly computing the total time derivative of u_1 and u_2 and writing

$$\dot{u_1} = \mathbf{L}_1^\top \mathbf{v} \text{ and } \dot{u_2} = \mathbf{L}_2^\top \mathbf{v}, \tag{20}$$

we obtain the interaction matrix related to the intensity at pixel $\mathbf{x}$ in the general case

$$\mathbf{L}_I = -\boldsymbol{\nabla} I^\top \mathbf{L}_\mathbf{x} + kK_s {u_1}^{k-1}\mathbf{L}_1^\top + K_d \mathbf{L}_2^\top. \tag{21}$$

Note that we recover here the interaction matrix $-\boldsymbol{\nabla} I^\top \mathbf{L}_\mathbf{x}$ associated to the intensity under temporal constancy (see (9)), i.e. in the Lambertian case ($K_s = 0$) and when $\dot{u_2} = 0$ (i.e. the lighting direction is motionless with respect to the point $\mathbf{P}$).

We now compute the vectors $\mathbf{L}_1$ and $\mathbf{L}_2$ involved in (20) to explicitly compute $\mathbf{L}_I$.

2.2.1 Computation of $\mathbf{L}_2$

This computation requires to write

$$\dot{u_2} = \mathbf{L}^\top \dot{\mathbf{n}} + \mathbf{n}^\top \dot{\mathbf{L}}. \tag{22}$$

In the general case of the time dependences given by (16), (22) becomes

$$\begin{aligned} \dot{u_2} &= \mathbf{L}^\top \left(\mathbf{J}^\mathbf{n} \dot{\mathbf{x}} + \frac{\partial \mathbf{n}}{\partial t} \right) + \mathbf{n}^\top \left(\mathbf{J}^\mathbf{L} \dot{\mathbf{x}} + \frac{\partial \mathbf{L}}{\partial t} \right) && (23) \\ &= \left(\mathbf{L}^\top \mathbf{J}^\mathbf{n} + \mathbf{n}^\top \mathbf{J}^\mathbf{L} \right) \dot{\mathbf{x}} + \mathbf{L}^\top \frac{\partial \mathbf{n}}{\partial t} + \mathbf{n}^\top \frac{\partial \mathbf{L}}{\partial t} && (24) \end{aligned}$$

where $\mathbf{J}^\mathbf{n}$ and $\mathbf{J}^\mathbf{L}$ express respectively the Jacobian matrices related to $\mathbf{n}$ and $\mathbf{L}$ with respect to $\mathbf{x}$.

However, if we want to express $\partial \mathbf{n}/\partial t$ and $\partial \mathbf{L}/\partial t$ in function of the camera velocity $\mathbf{v}$, we have to do some hypothesis about how $\mathbf{n}$ and $\mathbf{L}$ move with respect to the observer. We consider two cases.

• *Light source motionless with respect to the object frame.* To compute $\partial \mathbf{n}/\partial t$, we introduce the matrix ${}^c\mathbf{R}_o$ which describes the rotation between the camera and the object frames such that $\mathbf{n} = {}^c\mathbf{R}_o {}^o\mathbf{n}$ where ${}^o\mathbf{n}$ expresses $\mathbf{n}$ in the object frame. Therefore,

$$\frac{\partial \mathbf{n}}{\partial t} = {}^c\dot{\mathbf{R}}_o {}^o\mathbf{n} = {}^c\dot{\mathbf{R}}_o {}^c\mathbf{R}_o^\top \mathbf{n} = -\boldsymbol{\omega} \times \mathbf{n}. \tag{25}$$

Note that we similarly have

$$\frac{\partial \mathbf{L}}{\partial t} = -\boldsymbol{\omega} \times \mathbf{L}. \tag{26}$$

In that case, it is straightforward to show that

$$\begin{aligned} \frac{\partial u_2}{\partial t} &= \mathbf{L}^\top \frac{\partial \mathbf{n}}{\partial t} + \mathbf{n}^\top \frac{\partial \mathbf{L}}{\partial t} && (27) \\ &= 0 && (28) \end{aligned}$$

which directly gives from (24)

$$\mathbf{L}_2^\top = \left(\mathbf{L}^\top \mathbf{J}^\mathbf{n} + \mathbf{n}^\top \mathbf{J}^\mathbf{L} \right) \mathbf{L}_\mathbf{x} \tag{29}$$

where we have introduced the interaction matrix related to $\mathbf{x}$.

• *Light source mounted on the camera.* We thus have $\mathbf{L} = -\mathbf{k}$. In that case, the equations become simpler since we have $\mathbf{J}^{\mathbf{L}} = \mathbf{0}$ and $\partial \mathbf{L}/\partial t = \mathbf{0}$.

Let us rewrite $\dot{u_2}$ from (24) where we have introduced $\mathbf{L_x}$ and (25)

$$\begin{aligned} \dot{u_2} &= -\left(\mathbf{k}^\top \mathbf{J}^{\mathbf{n}}\right) \mathbf{L_x v} - \mathbf{k}^\top \left(\mathbf{n} \times \boldsymbol{\omega}\right) & (30) \\ &= -\boldsymbol{\nabla} n_z^\top \mathbf{L_x v} + \left(\mathbf{n} \times \mathbf{k}\right)^\top \boldsymbol{\omega}. & (31) \end{aligned}$$

Therefore, by introducing the following vector

$$\mathbf{L}_4^\top = \begin{pmatrix} 0 & 0 & 0 & (\mathbf{n} \times \mathbf{k})^\top \mathbf{i} & (\mathbf{n} \times \mathbf{k})^\top \mathbf{j} & 0 \end{pmatrix}, \tag{32}$$

$\mathbf{L}_2^\top$ expresses as follows

$$\mathbf{L}_2^\top = -\boldsymbol{\nabla} n_z^\top \mathbf{L_x} + \mathbf{L}_4^\top. \tag{33}$$

2.2.2 Computation of $\mathbf{L}_1$

Let us recall that $u_1 = \mathbf{R}^\top \mathbf{V}$ leading, by considering the time dependences given in (16), to

$$\dot{u_1} = \mathbf{V}^\top \dot{\mathbf{R}} + \mathbf{R}^\top \dot{\mathbf{V}} \tag{34}$$

which gives

$$\begin{aligned} \dot{u_1} &= \mathbf{V}^\top \left(\mathbf{J}^{\mathbf{R}} \dot{\mathbf{x}} + \frac{\partial \mathbf{R}}{\partial t} \right) + \mathbf{R}^\top \mathbf{J}^{\mathbf{V}} \dot{\mathbf{x}} & (35) \\ &= \left(\mathbf{V}^\top \mathbf{J}^{\mathbf{R}} + \mathbf{R}^\top \mathbf{J}^{\mathbf{V}} \right) \dot{\mathbf{x}} + \mathbf{V}^\top \frac{\partial \mathbf{R}}{\partial t} & (36) \end{aligned}$$

where $\mathbf{J}^{\mathbf{R}}$ and $\mathbf{J}^{\mathbf{V}}$ express respectively the Jacobian matrices related to $\mathbf{R}$ and $\mathbf{V}$ with respect to $\mathbf{x}$. In addition, since $\mathbf{R}$ is a function of $\mathbf{L}$ and $\mathbf{n}$ (see (15)), we have

$$\frac{\partial \mathbf{R}}{\partial t} = 2 \left(\frac{\partial u_2}{\partial t} \mathbf{n} + u_2 \frac{\partial \mathbf{n}}{\partial t} \right) - \frac{\partial \mathbf{L}}{\partial t} \tag{37}$$

and after some manipulations detailed in appendix

$$\mathbf{J}^{\mathbf{R}} = 2\mathbf{n}\left(\mathbf{L}^\top \mathbf{J}^{\mathbf{n}} + \mathbf{n}^\top \mathbf{J}^{\mathbf{L}}\right) + 2u_2 \mathbf{J}^{\mathbf{n}} - \mathbf{J}^{\mathbf{L}}. \tag{38}$$

The computation of $\mathbf{J}^{\mathbf{V}}$ is quite simple, it is given by

$$\mathbf{J}^{\mathbf{V}} = \frac{1}{\|\mathbf{x}\|^3} \begin{bmatrix} -(1+y^2) & xy \\ xy & -(1+x^2) \\ x & y \end{bmatrix}. \tag{39}$$

At this step, as we did for the computation of $\mathbf{L}_2$, we consider the cases when the light source is motionless with respect to the object frame or when the light source is mounted on the camera.

• *Light source motionless with respect to the object frame.* From (27), (25) and (26), (37) becomes

$$\frac{\partial \mathbf{R}}{\partial t} = 2u_2(\mathbf{n} \times \boldsymbol{\omega}) - (\mathbf{L} \times \boldsymbol{\omega}) = \mathbf{R} \times \boldsymbol{\omega} \tag{40}$$

and by introducing the vector $\mathbf{L}_3^\top$ such as

$$\begin{aligned} \mathbf{V}^\top \frac{\partial \mathbf{R}}{\partial t} &= \mathbf{L}_3^\top \mathbf{v} & (41) \\ &= \begin{pmatrix} 0 & 0 & 0 & \mathbf{W}^\top \mathbf{i} & \mathbf{W}^\top \mathbf{j} & \mathbf{W}^\top \mathbf{k} \end{pmatrix} \mathbf{v} & (42) \end{aligned}$$

with $\mathbf{W} = \mathbf{V} \times \mathbf{R}$, we obtain $\mathbf{L}_1^\top$ from (36) by introducing $\mathbf{L}_\mathbf{x}$

$$\mathbf{L}_1^\top = \big(\mathbf{V}^\top \mathbf{J}^\mathbf{R} + \mathbf{R}^\top \mathbf{J}^\mathbf{V}\big)\mathbf{L}_\mathbf{x} + \mathbf{L}_3^\top. \tag{43}$$

• *Light source mounted on the camera.* Recall that we have in this case $\mathbf{L} = -\mathbf{k}$ leading to $\mathbf{J}^\mathbf{L} = \mathbf{0}$ and $\partial \mathbf{L}/\partial t = \mathbf{0}$.

To rewrite u_1 given in (36), we first compute (37) using $u_2 = -\mathbf{k}^\top \mathbf{n}$

$$\frac{\partial \mathbf{R}}{\partial t} = -2\left(\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial t}\mathbf{n} + \mathbf{k}^\top \mathbf{n}\frac{\partial \mathbf{n}}{\partial t}\right) \tag{44}$$

leading by using (25) to

$$\begin{aligned}
\mathbf{V}^\top \frac{\partial \mathbf{R}}{\partial t} &= -2\big(\mathbf{k}^\top \big(\mathbf{n} \times \boldsymbol{\omega}\big)\mathbf{V}^\top \mathbf{n} + \mathbf{k}^\top \mathbf{n} \mathbf{V}^\top \big(\mathbf{n} \times \boldsymbol{\omega}\big)\big) &(45)\\
&= -2\big(\mathbf{n}^\top \mathbf{V}\big(\mathbf{k} \times \mathbf{n}\big)^\top \boldsymbol{\omega} + \mathbf{k}^\top \mathbf{n}\big(\mathbf{V} \times \mathbf{n}\big)^\top \boldsymbol{\omega}\big). &(46)
\end{aligned}$$

Consequently, by introducing the vector $\mathbf{L}_5^\top$ such as

$$\begin{aligned}
\mathbf{V}^\top \frac{\partial \mathbf{R}}{\partial t} &= \mathbf{L}_5^\top \mathbf{v} &(47)\\
&= \begin{pmatrix} 0 & 0 & 0 & \mathbf{L}_{5x} & \mathbf{L}_{5y} & \mathbf{L}_{5z} \end{pmatrix} \mathbf{v} &(48)
\end{aligned}$$

with

$$\left\{\begin{array}{lcl}
\mathbf{L}_{5x} &=& 2\big(\mathbf{n}^\top \mathbf{V}(\mathbf{n} \times \mathbf{k})^\top + \mathbf{k}^\top \mathbf{n}\big(\mathbf{n} \times \mathbf{V}\big)^\top\big)\mathbf{i} \\
\mathbf{L}_{5y} &=& 2\big(\mathbf{n}^\top \mathbf{V}(\mathbf{n} \times \mathbf{k})^\top + \mathbf{k}^\top \mathbf{n}\big(\mathbf{n} \times \mathbf{V}\big)^\top\big)\mathbf{j} \\
\mathbf{L}_{5z} &=& 2\mathbf{k}^\top \mathbf{n}\big(\mathbf{n} \times \mathbf{V}\big)^\top \mathbf{k}
\end{array}\right. \tag{49}$$

we obtain from (36)

$$\mathbf{L}_1^\top = \big(\mathbf{V}^\top \mathbf{J}^\mathbf{R} + \mathbf{R}^\top \mathbf{J}^\mathbf{V}\big)\mathbf{L}_\mathbf{x} + \mathbf{L}_5^\top. \tag{50}$$

Let us point out that $\mathbf{J}^\mathbf{R}$ becomes also simpler

$$\begin{aligned}
\mathbf{J}^\mathbf{R} &= 2\mathbf{n}\big(\mathbf{L}^\top \mathbf{J}^\mathbf{n}\big) + 2u_2 \mathbf{J}^\mathbf{n} &(51)\\
&= -2\big(\mathbf{n}\big(\mathbf{k}^\top \mathbf{J}^\mathbf{n}\big) + \mathbf{n}^\top \mathbf{k} \mathbf{J}^\mathbf{n}\big) &(52)\\
&= -2\big(\mathbf{n} \boldsymbol{\nabla} n_z^\top + n_z \mathbf{J}^\mathbf{n}\big). &(53)
\end{aligned}$$

3 Interaction matrix in some particular cases

In classical visual servoing the interaction matrix is very often computed at the desired position [1]. This way to proceed avoid to compute on-line 3D information like the depths for example. We also consider this case in this section. More precisely, we consider that, at the desired position the depth of all the points where the luminance is measured are equal to a constant value Z^*. That means that we consider that the object is planar and that the camera and the object planes are parallel at this position.

3.1 Light source motionless with respect to the object frame and located at infinity

This case is depicted on figure 2. Since the object is planar, $\mathbf{n}$ does no longer depend on $\mathbf{x}$, then $\mathbf{J^n} = \mathbf{0}$. Similarly, since the light source is at infinity, $\mathbf{L}$ does not depend on $\mathbf{x}$ and thus $\mathbf{J^L} = \mathbf{0}$. In addition, since the angle between $\mathbf{n}$ and $\mathbf{L}$ is constant, $u_2 = \mathbf{n}^\top \mathbf{L}$ is constant.

Consequently, it is easy to show from (29) that $\mathbf{L}_2^\top = \mathbf{0}$.

For $\mathbf{L}_1^\top$, since $\mathbf{J^n} = \mathbf{J^L} = \mathbf{0}$, we have also $\mathbf{J^R} = \mathbf{0}$ (see (38)), leading from (43) to

$$\mathbf{L}_1^\top = \mathbf{R}^\top \mathbf{J^V} \mathbf{L_x} + \mathbf{L}_3^\top. \tag{54}$$

Thus, $\mathbf{L}_1^\top$ can be easily computed. All computations done, we obtain

$$\mathbf{L}_1^\top = \frac{1}{Z\|\mathbf{x}\|^3} \begin{pmatrix} L_{1x} & L_{1y} & L_{1z} & 0 & 0 & 0 \end{pmatrix} \tag{55}$$

with

$$\left\{ \begin{array}{lcl} L_{1x} & = & \left(1+y^2\right) R_x - xyR_y - xR_z \\ L_{1y} & = & -xyR_x + \left(1+x^2\right) Ry - yR_z \\ L_{1z} & = & -xR_x - yR_y + \left(x^2+y^2\right) R_z \end{array} \right. \tag{56}$$

where R_x, R_y, R_z are the components of $\mathbf{R}$. Note that to compute $\mathbf{R}$ the pose of the camera with respect to the object is required. However, if we consider the particular case where the camera and the object plane are parallel, we have $\mathbf{n} = -\mathbf{k}$ which leads to

$$\mathbf{R} = -2u_2\mathbf{k} - L \tag{57}$$

that can be easily evaluated.

As can be seen, even if the computation of the vectors $\mathbf{L}_1$ and $\mathbf{L}_2$ to derive the interaction matrix is not straightforward, their final expression is very simple and easy to compute.

3.2 Interaction matrix when the light source is mounted on the camera and computed at the desired position

This case is depicted on the figure 3. Here, since $\mathbf{J^n} = \mathbf{0}$ and $\mathbf{n} = -\mathbf{k}$, from (32) and (33), it is straightforward to show that $\mathbf{L}_2^\top = \mathbf{0}$. Besides, since $\mathbf{n} = -\mathbf{k}$ and $\mathbf{L} = -\mathbf{k}$, we have $\mathbf{R} = -\mathbf{k}$. We also have $\mathbf{J^R} = \mathbf{0}$. Consequently, from (50), $\mathbf{L}_1^\top$ becomes

$$\mathbf{L}_1^\top = -\mathbf{k}^\top \mathbf{J^V} \mathbf{L_x} + \mathbf{L}_3^\top \tag{58}$$

while $\mathbf{L}_3^\top$ becomes from (49)

$$\mathbf{L}_3^\top = \begin{pmatrix} 0 & 0 & 0 & -2\mathbf{V}^\top\mathbf{j} & -2\mathbf{V}^\top\mathbf{i} & 0 \end{pmatrix}. \tag{59}$$

Using explicitly $\mathbf{V}$, $\mathbf{J^V}$ and $\mathbf{L_x}$, we simply obtain

$$\mathbf{L}_1^\top = \frac{1}{\|\mathbf{x}\|} \begin{pmatrix} \frac{x}{\bar{Z}} & \frac{y}{\bar{Z}} & -\frac{x^2+y^2}{\bar{Z}} & y & -x & 0 \end{pmatrix} \tag{60}$$

where $\bar{Z} = Z^*\| \mathbf{x} \|^2$.

Here again, the final expression of $\mathbf{L}_1$ and $\mathbf{L}_2$ is very simple and easy to compute.

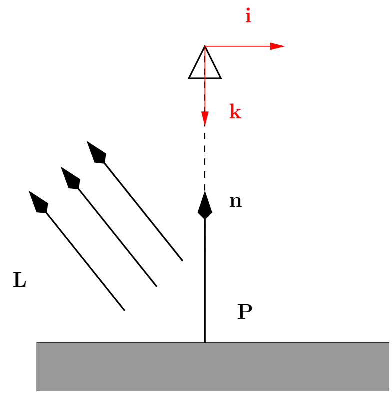


Figure 2: Light source located at infinity and camera and object planes parallel.

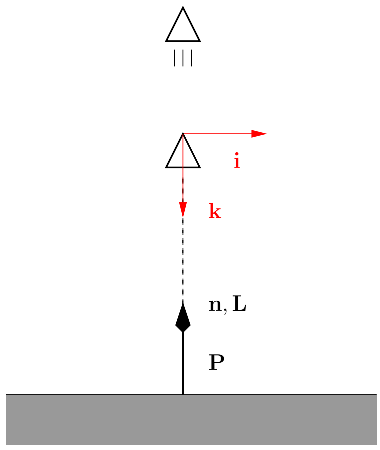


Figure 3: Light source mounted on the camera for a planar object when the camera and the object planes are parallel.

4 Visual servoing control law

Since the interaction matrix associated to the luminance is now known, the control law can be derived. However, we turn here the visual servoing problem into an optimization problem as proposed in [33].

4.1 Visual servoing as an optimization problem

Different control laws can be derived regarding the minimization technique one uses. Let us recall that the goal is to minimize the following cost function

$$\mathcal{C}(\mathbf{r}) = \frac{1}{2} \parallel \mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*) \parallel^2 \tag{61}$$

where $\mathbf{r}$ describes the current pose of the camera with respect to the object (it is an element of $\mathbb{R}^3 \times SO(3)$) and where $\mathbf{r}^*$ is the desired pose. Several methods are detailed in [33], we only give here the most interesting results while focusing on the differential approaches. In that case, a step of the minimization scheme can be written as follows

$$\mathbf{r}_{k+1} = \mathbf{r}_k \oplus t_k \mathbf{d}\left(\mathbf{r_k}\right) \tag{62}$$

where "$\oplus$" denotes the operator that combines two consecutive frame transformations, $\mathbf{r}_k$ is the current pose, t_k is a positive scalar (the descent step) and $\mathbf{d}\left(\mathbf{r_k}\right)$ a direction of descent ensuring that (61) decreases if

$$\mathbf{d}\left(\mathbf{r_k}\right)^\top \nabla \mathcal{C}\left(\mathbf{r}_k\right) < 0. \tag{63}$$

In that case, the following velocity control law can be derived considering that t_k is small enough

$$\mathbf{v} = \lambda_k \mathbf{d}\left(\mathbf{r_k}\right) \tag{64}$$

where λ_k is a scalar that depends on t_k and on the sampling rate. It is often chosen as a constant value. In the remainder of the report we will omit the subscript k for the sake of clarity.

4.1.1 Steepest descent (gradient method)

The direction of descent (used for example in [34]) is simply

$$\mathbf{d}\left(\mathbf{r}\right) = -\nabla \mathcal{C}\left(\mathbf{r}\right) \tag{65}$$

where

$$\nabla \mathcal{C}\left(\mathbf{r}\right) = \left(\frac{\partial \mathbf{I}}{\partial \mathbf{r}}\right)^\top \left(\mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)\right). \tag{66}$$

Since we have $\dot{\mathbf{I}} = \frac{\partial \mathbf{I}}{\partial \mathbf{r}}\dot{\mathbf{r}} = \mathbf{L_I}\mathbf{v}$, we obtain the following control law

$$\mathbf{v} = -\lambda \mathbf{L_I}^\top \big(\mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)\big). \tag{67}$$

4.1.2 Gauss-Newton

When $\mathbf{r}_k$ lies in a neighborhood of $\mathbf{r}^*$, $\mathbf{I}(\mathbf{r})$ can be linearized around $\mathbf{I}(\mathbf{r_k})$ and plugged into (61). Then, after having zeroed its gradient, we obtain

$$\mathbf{d}\left(\mathbf{r}\right) = -\left(\left(\frac{\partial \mathbf{I}}{\partial \mathbf{r}}\right)^\top \left(\frac{\partial \mathbf{I}}{\partial \mathbf{r}}\right)\right)^{-1} \nabla \mathcal{C}\left(\mathbf{r}\right) \tag{68}$$

that becomes using (66)

$$\mathbf{v} = -\lambda \mathbf{L_I}^+ \big(\mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)\big) \tag{69}$$

which is nothing but (2), that is the control law usually used.

4.1.3 Newton

If we locally approximate $\mathcal{C}(\mathbf{r})$ by its second order Taylor series expansion in $\mathbf{r}_k$ and cancel its gradient, we have

$$\mathbf{d}\left(\mathbf{r}\right) = -\big(\nabla^2 \mathcal{C}(\mathbf{r})\big)^{-1} \nabla \mathcal{C}(\mathbf{r}) \tag{70}$$

with

$$\nabla^2 \mathcal{C}(\mathbf{r}) = \left(\frac{\partial \mathbf{I}}{\partial \mathbf{r}}\right)^\top \left(\frac{\partial \mathbf{I}}{\partial \mathbf{r}}\right) + \sum_{i=1}^{i=\dim \mathbf{I}} \nabla^2 s_i \left(I_i(\mathbf{r}) - I_i(\mathbf{r}^*)\right). \tag{71}$$

This approach has been considered in [35] for example. Note that the vector $\mathbf{d}(\mathbf{r})$ is really a direction of descent if $\nabla^2\mathcal{C}(\mathbf{r}) > 0$ holds (see (63)). Note also that the Newton's and Gauss-Newton's approaches are equivalent in $\mathbf{r}^*$.

4.1.4 Levenberg-Marquardt

This method considers the following direction

$$\mathbf{d}(\mathbf{r}) = -\big(\mathbf{G} + \mu \text{ diag}(\mathbf{G})\big)^{-1} \nabla\mathcal{C}(\mathbf{r}) \tag{72}$$

where $\mathbf{G}$ is usually chosen as $\nabla^2\mathcal{C}(\mathbf{r})$ or more simply as $\left(\frac{\partial \mathbf{I}}{\partial \mathbf{r}}\right)^\top \left(\frac{\partial \mathbf{I}}{\partial \mathbf{r}}\right)$ leading in that last case to

$$\mathbf{v} = -\lambda\big(\mathbf{H} + \mu \text{ diag}(\mathbf{H})\big)^{-1} \mathbf{L}_\mathbf{I}^\top \big(\mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)\big) \tag{73}$$

with $\mathbf{H} = \mathbf{L_I}^\top \mathbf{L_I}$. The parameter μ makes possible to switch from a steepest descent like approach to a Gauss-Newton one thanks to the observation of (61) during the minimization process. Indeed, when μ is very high (73) behaves like (67) [2]. In contrast, when μ is very low (73) behaves like (69).

4.1.5 ESM

In [33] a second order method which does not required the computation of $\nabla^2\mathcal{C}(\mathbf{r})$ is presented. Indeed, we have

$$\mathbf{v} = -\lambda\big(\mathbf{L_I} + \mathbf{L_{I^*}}\big)^+ \big(\mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)\big). \tag{74}$$

In contrast with the classical minimization algorithms, this approach takes benefit from the behavior of the cost function that is known in the neighborhood of the minimum.

4.2 Analysis of the cost function

Since the convergence of the control laws described in Section 4.1 highly depends on the cost function (61), we focus here on its shape.

To do that, we consider the vector $\mathbf{I}(\mathbf{r})$ given by (3). We write $\mathbf{r} = (\mathbf{t}, \mathbf{u}\theta)$ where $\mathbf{t} = (t_x, t_y, t_z)$ describes the translation part of the homogeneous matrix related to the transformation from the current to the desired frame, while its rotation part is expressed under the form $\mathbf{u}\theta$ where $\mathbf{u}$ represents the unit rotation axis vector and θ the rotation angle around this axis.

As an example, Fig. 4b, e, h, k, n and Fig. 4c, f, i, l and o describe the shape of cost functions (61) in the subspace (t_x, θ_y) when the scene being observed is planar (see the figures 4a, d, g, j and m) and when the desired pose is such that the image plane and the object plane are parallel at the depth Z^* = 80 cm. Let us point out that this is the most complex case (with its dual case (t_y, θ_x)). Indeed, it is well known that it is very difficult to distinguish in an image an x axis translational motion (respectively y) from

[2] More precisely, each component of the gradient is scaled according to the diagonal of the Hessian, which leads to larger displacements along the direction where the gradient is low.

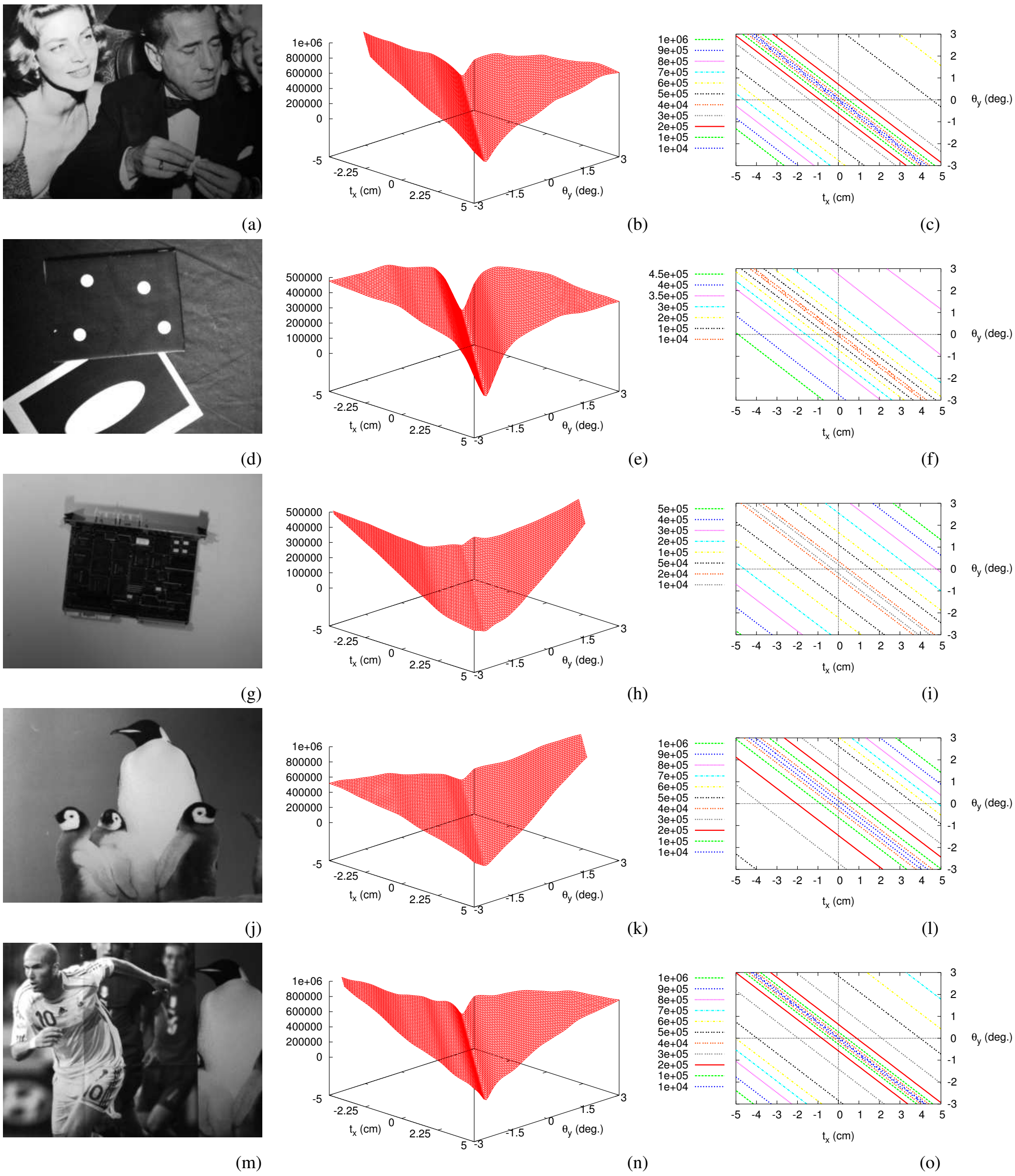


Figure 4: Cost function for different objects. First column: object being observed, second column: shape of the cost function in the subspace (t_x, θ_y), third column: iso-contours in the subspace (t_x, θ_y). There is always a narrow valley at the middle of a gentle slope plateau with non constant slope.

a y axis rotational motion (respectively x). It explains why the cost function is low in a preferential direction, as clearly shown on Fig. 4b, e, h, k, n and Fig. 4c, f, i, l and o. In addition, the cost functions highly depends on the pose $\mathbf{r}$ since they rapidly increase after their minimum (see Fig. 4b, e, h, k and n). Moreover, as can be seen, the shape of the cost functions (61) does not depend too much on the scene content as soon as the image does not contain periodic patterns or strong changes of the spatial gradient. It always shows a narrow valley at the middle of a gentle slope plateau with non constant slope. Note that (61) is only quasi convex, moreover on a very small domain.

Let us study more precisely (61) in a neighborhood of $\mathbf{r}^*$. To do that, we perform a first order Taylor series expansion of the visual features $\mathbf{I}(\mathbf{r})$ around $\mathbf{r}^*$ by

$$\mathbf{I}(\mathbf{r}) = \mathbf{I}(\mathbf{r}^*) + \mathbf{L}_{\mathbf{I}^*}\mathbf{\Delta r} \tag{75}$$

where $\mathbf{\Delta r}$ denotes the relative pose between $\mathbf{r}$ and $\mathbf{r}^*$. Therefore, by plugging (75) into (61), the cost function can be approximated in a neighborhood of $\mathbf{r}^*$

$$\widehat{\mathcal{C}(\mathbf{r})} = \mathbf{\Delta r}^\top \mathbf{H}^* \mathbf{\Delta r} \tag{76}$$

with $\mathbf{H}^* = \mathbf{L}_{\mathbf{I}^*}^\top \mathbf{L}_{\mathbf{I}^*}$. Due to the complexity of the interaction matrix, we restrict this study to the Lambertian case. In practice, because of the special form of the interaction matrix given in (9) (its translation part contains terms related to the depths), the eigenvalues of the matrix $\mathbf{H}^*$ are very different (unfortunately, only numerical results can be obtained because of the complexity of this matrix). This result also holds for most of the geometrical visual features where a term related to the depth occurs in the translational part of the interaction matrix. Consequently, in the subspace (t_x, θ_y) (respectively (t_y, θ_x)), the cost function is an elliptic paraboloid with a very high major axis with respect to its minor axis leading consequently to near parallel isocontours as shown on Fig. 4c, f, i, l and o. Moreover, the eigenvectors of $\mathbf{H}^*$ point out some directions where the cost function decreases slowly when its associated eigenvalue is low or decreases quickly when its associated eigenvalue is high. In the case of Fig. 4c, f, i, l and o, the eigenvector associated to the smaller eigenvalue corresponds to the valley where the cost varies slowly. In contrast, it varies strongly along an orthogonal direction, that is in a direction near $\nabla\mathcal{C}(\mathbf{r})$. We will use this knowledge about the cost function in the next section to derive an efficient control law.

4.3 Positioning tasks

As shown in Section 4.1, several control laws can be used to minimize (61). We first used the classical control laws based on the Gauss-Newton approach and the ESM approach [33, 36]. Unfortunately, they all failed, either because they diverged or because they led to unsuitable 3D motion. Therefore, a new control law has to be derived.

Indeed, since the general form of the cost function is known (see Fig. 4b, e, h, k and n), we propose the following algorithm to reach its minimum. The camera is first moved to reach the valleys and next along the axes of the valleys towards the desired pose. The first step can be easily done by using a gradient approach. However, as seen on Fig. 4c, f, i, l and o, the direction of $\nabla\mathcal{C}(\mathbf{r})$ is constant but its amplitude on the plateau is not constant (see Fig. 4b, e, h, k and n) since the slope varies. We could tune the parameter λ involved in (67) to ensure smooth 3D velocities. However, a simpler approach to achieve this goal consists in using the following control law

$$\mathbf{v} = -v_c \frac{\nabla\mathcal{C}(\mathbf{r}_{init})}{\| \nabla\mathcal{C}(\mathbf{r}_{init}) \|}. \tag{77}$$

That is, a constant velocity with norm v_c is applied in the steepest descent computed at the initial camera pose. Consequently, this first step behaves as an open-loop system. To turn into a closed-loop system, we first detect roughly the bottom of the valley from a 3rd order polynomial filtering of $\mathcal{C}(\mathbf{r})$ and then apply the control law (73). In addition, rather to control the parameter μ as in the Levenberg-Marquardt algorithm, a different way to proceed is used as detailed below. We denote MLM this method in the remainder of the report. Instead of using for the matrix $\mathbf{H}$ the Hessian of the cost function, we use its approximation $\mathbf{L_I}^\top \mathbf{L_I}$. The resulting control law is then given by

$$\mathbf{v} = -\lambda \left(\mathbf{H} + \mu \operatorname{diag}(\mathbf{H})\right)^{-1} \mathbf{L_I}^\top \left(\mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)\right) \tag{78}$$

with $\mathbf{H} = \mathbf{L_I}^\top \mathbf{L_I}$.

We now detail how μ is tuned. Fig. 5a shows the paths obtained with the MLM algorithm in the case where $\mathbf{r}_{init} = (8$ cm, 4 cm, -10 cm, 3°, -3°, -5°$)$ for various choices of μ. If a high value is used, after the open-loop motion, the bottom of the valley is easily reached (see Fig. 5a when $\mu = 1$) since (78) behaves in this case like a steepest descent approach. But in this case, since the valley is narrow, the convergence rate towards the global minimum (following the direction of the axis of the valley) is very low (see Fig. 5b). In contrast, if μ is low, (78) behaves like a Gauss-Newton (GN) approach and the decrease of the cost function is faster but, the convergence is no more ensured (see the larger motion near the minimum on Fig. 5a when $\mu = 10^{-3}$). As can be seen, an intermediate value ($\mu = 10^{-2}$) has to be chosen to ensure a correct path (Fig. 5a) and a high convergence rate (Fig. 5b). Therefore, this value has been chosen in the experiments described in the next section.

5 Experimental results

In all the experiments reported here, the camera is mounted on a 6 degrees of freedom gantry robot. Control law is computed on a Core 2 Duo 3Gz PC running Linux. Image are acquired at 66Hz using an IEEE 1394 camera with a resolution of 320×240. The size of the vector $\mathbf{s}$ is then 76800. Despite this size, the interaction matrix $\mathbf{L_I}$ can be computed at each iteration if needed.

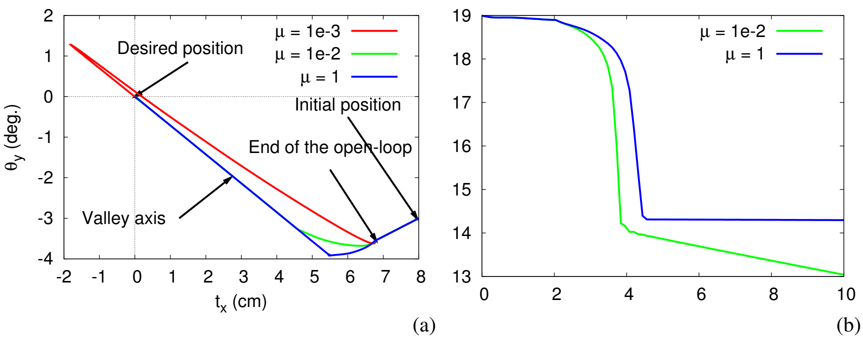


Figure 5: Influence of μ. (a) Path in the subspace (t_x, θ_y) for $\mathbf{r} = (8$ cm, 4 cm, -10 cm, 3°, -3°, -5°$)$, (b) Logarithm of the cost function versus time in second.

5.1 Positioning tasks under temporal luminance constancy

We assume in this section that the luminance $I(\mathbf{x})$ at a given pixel is constant. To make this assumption as valid as possible, a diffuse lighting as been used so that $I(\mathbf{x})$ can be considered as constant wrt to the viewing direction.

The goal of the first experiment is to compare the control laws based on GN and MLM approaches when the object described on Fig. 4a is considered. The initial error pose was $\mathbf{\Delta r}_{init}$ = (5 cm, -23 cm, 5 cm, -12.5°, -8.4°, -15.5°). The desired pose was so that the object and CCD planes are parallel at $Z^* = 80$ cm. The interaction matrix has been computed at each iteration but assuming that all the depths are constant and equal to Z^*, which is of course a coarse approximation.

Fig. 6a depicts the behavior of cost functions using the GN method or the MLM method while Fig. 6b depicts the trajectories (expressed in the desired frame) when using either the GN or the MLM method. Fig. 6c and Fig. 6d depict respectively the camera velocity. The initial and final images are reported respectively on Fig. 6e and Fig. 6f; First, as can be seen on Fig. 6a, both the control laws converge since the cost functions vanish. However, the time-to-convergence with the GN method is much higher than the one of the MLM method. The trajectory when using the GN method is also shaky compared to the one of the MLM method (Fig. 6b). The velocity of the camera when using the MLM method is smoother than when using the GN method (Fig. 6d and Fig. 6c). This experiment clearly shows that the MLM method outperforms the GN one. Note that in both cases the positioning errors is very low, for the MLM method we obtained $\mathbf{\Delta r}$ = (0.26 mm, 0.30 mm, 0.03 mm, 0.02°, -0.02°, 0.03°). It is very difficult to reach so low positioning errors when using geometric visual features. Indeed, these nice results are obtained because $\mathbf{I} - \mathbf{I}^*$ is very sensitive to the pose $\mathbf{r}$.

The goal of the next experiment is to show that, even if the luminance is used as a visual feature, our approach does not depend too much on the texture of the scene being observed. Fig. 7 depicts the behavior of our algorithm for the planar objects respectively given by Fig. 4d, g, j, m and p (the initial as well as the desired pose is unchanged). As can be seen, the control law converges in each cases, even in the case of a low textured scene (Fig. 4d and g). Let us point out that similar positioning errors than for the first experiment have been obtained.

The third experiment deals with partial occlusions. The desired object pose as well as the initial pose are still unchanged. After having moved the camera to its initial position, an object has been added to the scene, so that the initial image is now the one shown in Fig. 8a and the desired image is still the one shown in Fig. 6f. Moreover, the object introduced in the scene is also moved by hand, as seen in Fig. 8b and Fig. 8c, which highly increases the occluded surface. Despite that, the control law still converges (see Fig. 8f). Of course, since the desired image is not the true one, the error cannot vanish at the end of the motion (see Fig. 8f). Nevertheless, the positioning error is not affected by the occlusions since the final positioning error is $\mathbf{\Delta r}$ = (-0.1 mm, 2 mm, 0.3 mm, 0.13°, 0.04°, 0.07°) and it is very similar with the previous experiments. This very nice behavior is due to the high redundancy of the visual features we use.

The goal of the last experiment is to show the robustness of the control law wrt the depths. For this purpose, a non planar scene has been used as shown on Fig. 9. It shows that large errors in the depth are introduced (the height of the castle tower is around 30 cm). The initial and desired poses are unchanged. Fig. 10 depicts this experiment. Here again, the control law still converges (despite the interaction matrix

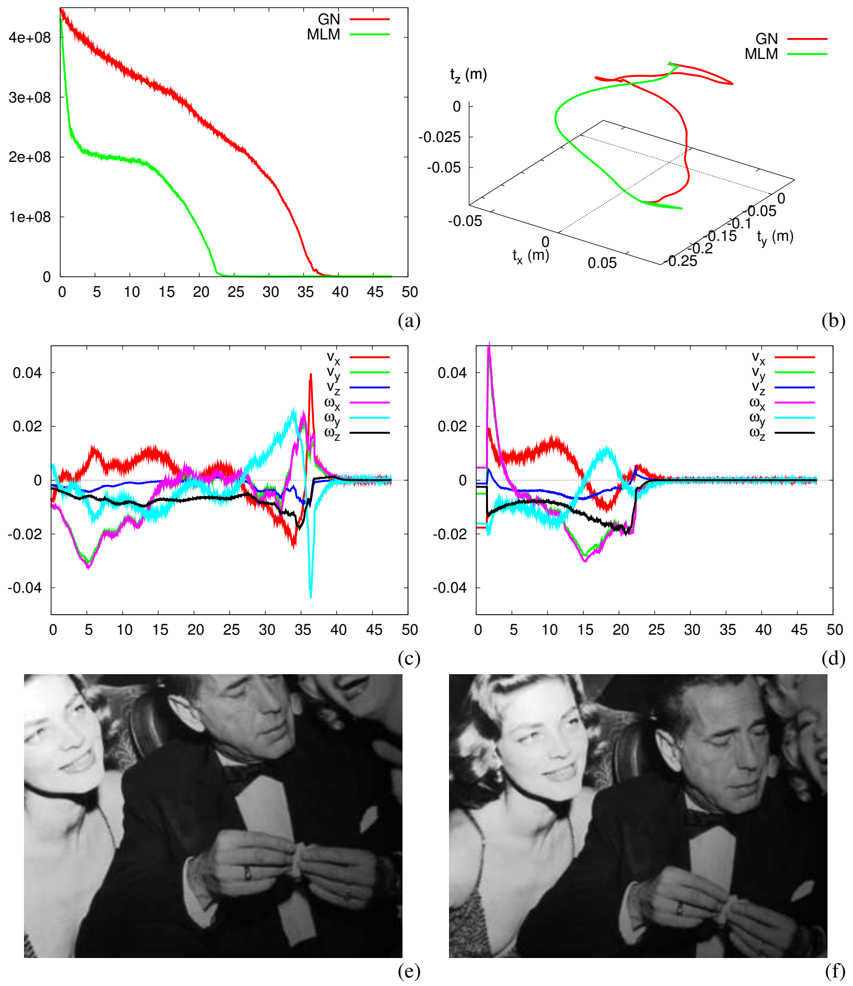


Figure 6: First experiment. MLM vs. GN method (x axis in second). (a) Comparison of cost functions, (b) Comparison of camera trajectories, (c) Camera velocities (m/s or rad/s) for the GN method, (d) Camera velocities (m/s or rad/s) for the MLM method, (e) Initial image, (f) Final image.

has been estimated at a constant depth $Z^* = 80$ cm) and the positioning error is still low since we have $\mathbf{\Delta r}$ = (0.2 mm, -0.0 mm, 0.1 mm, -0.01°, 0.00°, 0.06°).

5.2 Positioning tasks under complex illumination

In this section we consider the more complex case when the temporal luminance constancy luminance can no more be assumed.

5.2.1 Light source motionless with respect to the object frame and located at infinity

In this set of experiments a unique directional light has been added to the scene with a 45^o rotation around the vector $\mathbf{j}$ of the camera frame (see Fig. 2 for an illustration).

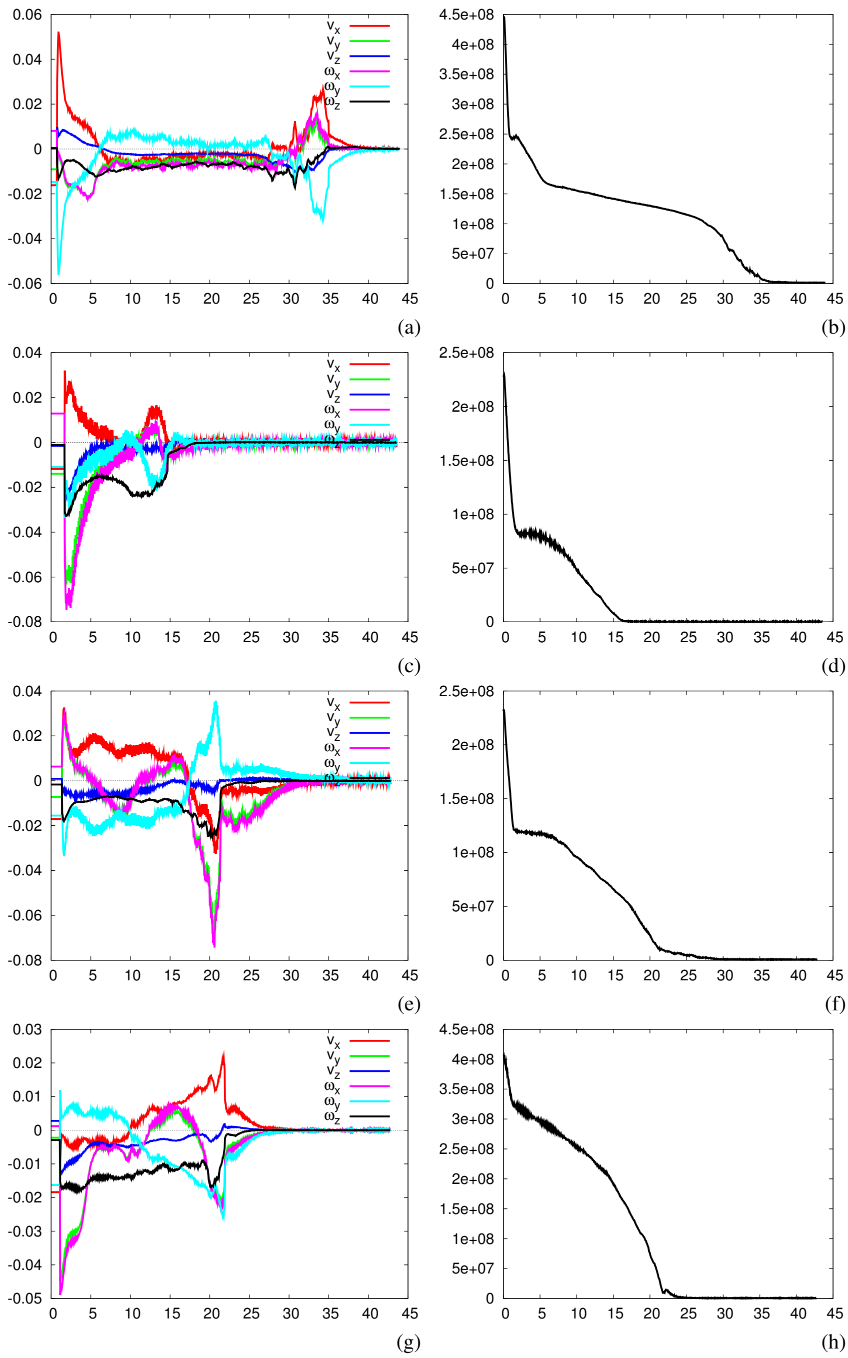


Figure 7: Second experiment (x axis in second). Behavior of the algorithm wrt to the objects respectively represented on Fig. 4d, g, j and m: (a), (c), (e) and (g) Camera velocities (m/s or rad/s), (b), (d), (f) and (h) Cost functions.

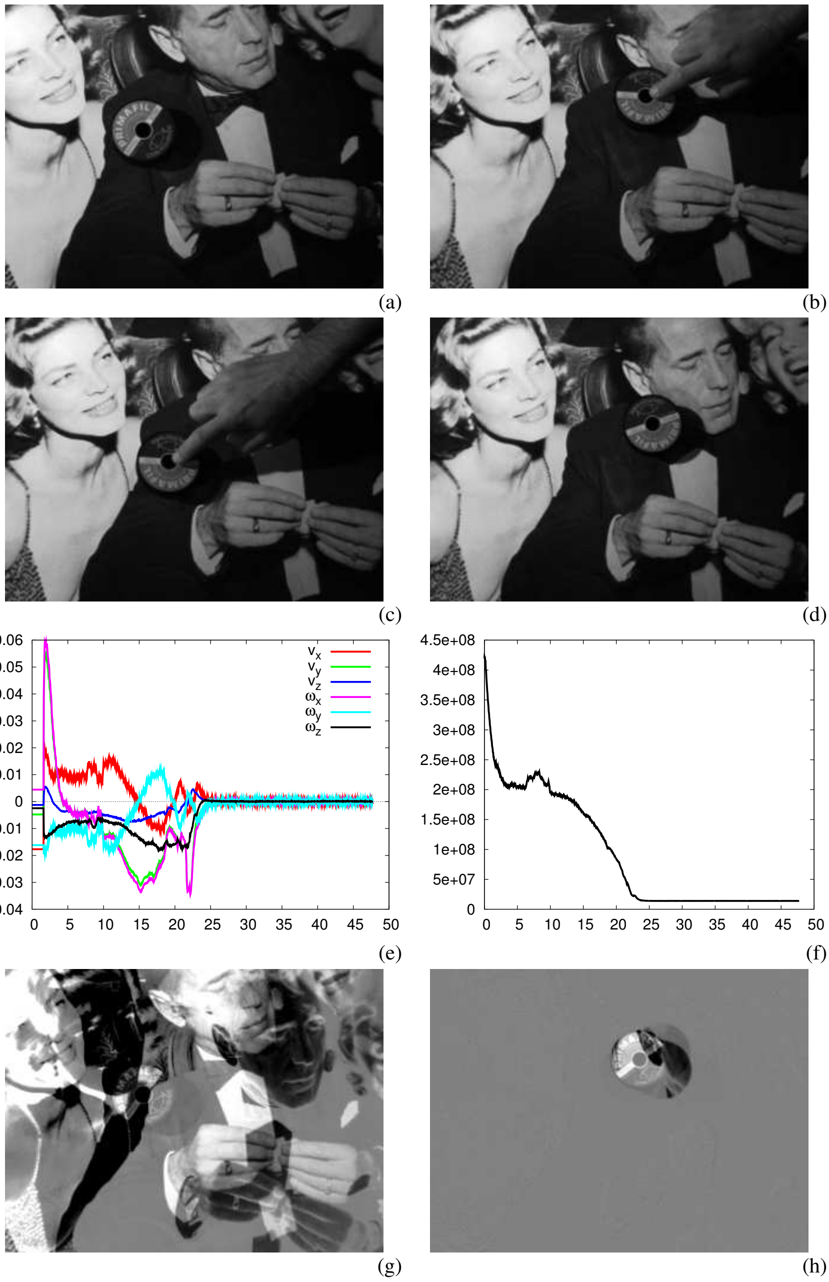


Figure 8: Third experiment. Occlusions (x axis in second). (a) Initial image, (b) Image at t $\approx$ 11 s, (c) Image at t $\approx$ 13 s (d) Final image, (e) Camera velocities (m/s or rad/s), (f) Cost function, (g) $\mathbf{I} - \mathbf{I}^*$ at the initial position, (h) $\mathbf{I} - \mathbf{I}^*$ at the end of the motion.

Figure 9: The non planar scene.

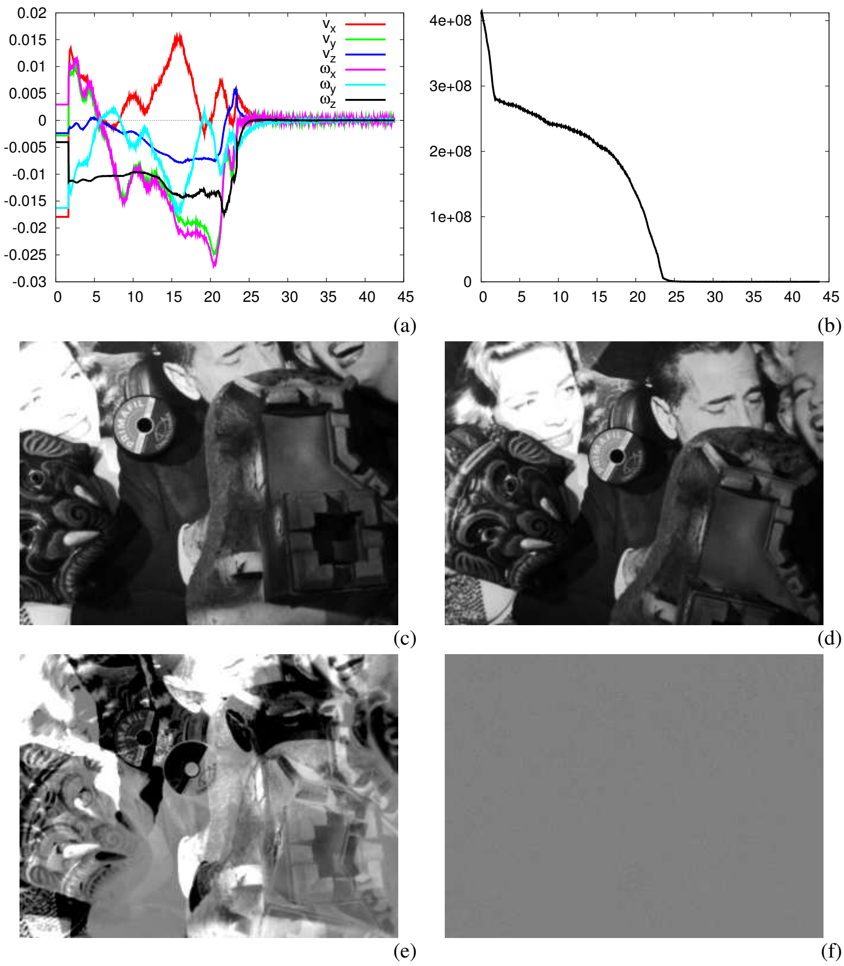


Figure 10: Fourth experiment. Robustness wrt depths (x axis in second). (a) Camera velocities (m/s or rad/s), (b) Cost function, (c) Initial image, (d) Final image, (g) $\mathbf{I} - \mathbf{I}^*$ at the initial position, (h) $\mathbf{I} - \mathbf{I}^*$ at the end of the motion.

This light produces an important specularity on the scene and then on the image (as can be shown on Figures 12b and 13b).

For this first experiment the initial positioning error was $\mathbf{\Delta r}$ = (-23 mm, -201 mm, 93 mm, -17.7°, 1.5°, -4.8°). First, let us note that in both cases (Lambertian model or the new proposed model) the robot converge toward the desired position (see Fig. 11). Nevertheless the convergence rate when considering explicitly specularities is always faster and smoother (this is due to a better estimation of the gradient of the cost function $\|\mathbf{e}\|^2$ that is used in the minimization process).

A similar experiment, although with a larger initial error, is shown on Figure 12. Indeed, the initial positioning error is $\mathbf{\Delta r}$ = (-24 mm, -176 mm, 86 mm, -13.75°, -6.76°, -30.53°). A similar behavior can be observed. Images shows the current images (images 0, 300 and 900) and the corresponding error $\mathbf{I} - \mathbf{I}^*$. Note that the specularity can be mainly seen near the head of the football player but as can be seen on the right images, it is in fact located all around the image. The final positioning error is still very low since we have $\mathbf{\Delta r} = (1 \text{ mm}, 0.04 \text{ mm}, 0 \text{ mm}, 0.04°, 0.08°, 0.1°)$.

The third experiment (see Fig. 13) describes the same experiment but with another and less textured scene. Here again a large specularity can be seen in the image. Despite these difficulties, the camera converges smoothly toward the desired position. The error $\mathbf{I} - \mathbf{I}^*$ is displayed on the top of Figure 13. The final positioning error is here again very low, we have $\mathbf{\Delta r} = (0 \text{ mm}, 0.02 \text{ mm}, 0.02 \text{ mm}, 0.01°, 0.01°, 0.05°)$.

5.2.2 Light source mounted on the camera

In this set of experiments a light-ring is located around the camera lens (see Fig. 14). Therefore the light direction is aligned with the camera optical axis as described on Figure 3 and thus it is moving with respect to the scene. This is the unique light in the scene. Note that, obviously, its direction is no more constant wrt the scene as previously. The initial positioning error and the desired pose are still unchanged (but with Z^*= 70cm). The interaction matrix has been estimated at the desired position using (60) to compute $\mathbf{L}_1^\top$ while $\mathbf{L}_2^\top = 0$ (see section 3.2). For all the experiments using the complete interaction matrix we used $k = 100$ and $K_s = 200$ (see (21)).

As can be seen on Fig. 15, the specularities are very important and consequently their motions in the image are important (for example the specularity can be seen at the bottom of the image in the first image whereas it has moved to the middle at the end of the positioning task). It also almost saturates the image meaning that few information are available around the specularity. The behavior of the robot is better since the convergence is faster and smoother when the complete model is considered (see Fig. 15a and Fig. 15.b and c).

5.3 Tracking tasks

Our goal is now to perform a tracking task with respect to a moving object. That is, we have to maintain a rigid link between the target to track and the camera. The lighting conditions are the same as those described in section 5.2.2 as well as the interaction matrix. However, the GN method has been used instead of the MLM method since the relative error pose between the desired and the current frames is considered as low. The object is still planar (a photo), and it is attached to a motorized rail that allows to control its motion (see Fig. 16). Although only one d.o.f of the object is controlled (with a motion that is completely unknown from the tracking process), the 6 d.o.f of the robot are controlled (the object velocity is 1 cm/s). Since we have a constant target

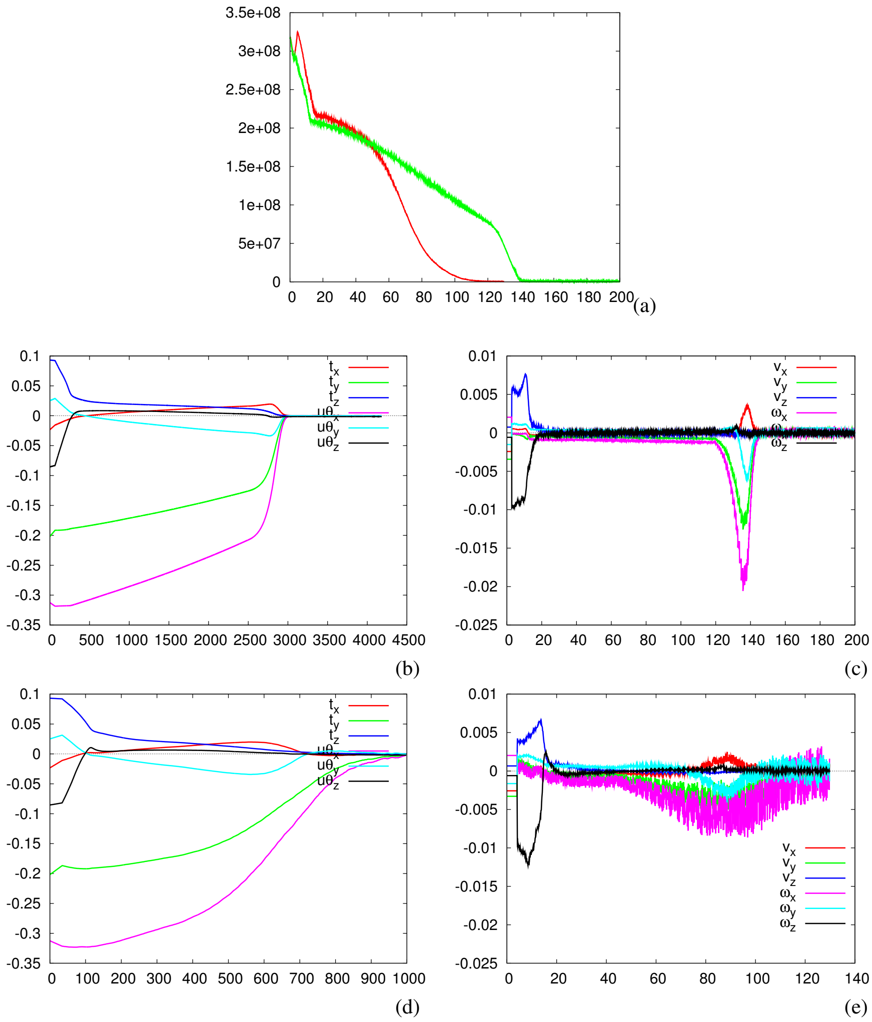
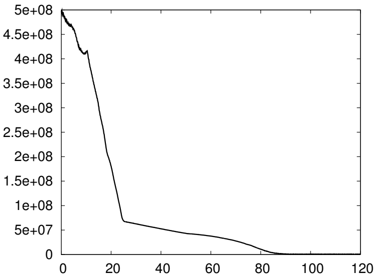


Figure 11: Positioning task with a light source motionless with respect to the object frame and located at "infinity" (x axis in second for (a), (c) and (e), frame number for (b) and (d)) . (a) cost function with a Lambertian model (green) and complete model (red), (b) positioning error (in m, rad) and (c) camera velocity for a Lambertian model, (d) positioning error (in m, rad) and (e) camera velocity for the complete model.

velocity, a simple integrator is considered in order to vanish the steady state tracking error as proposed in [37].

As can be seen on the images of Figures 17 (first rows) and 18.c, specularities can be seen in the image acquired by the camera. Figure 17 shows the behavior of the control law when the interaction matrix was computed only under temporal luminance constancy hypothesis (given by equation (9)). This figure shows that the tracking task quickly failed (the second row shows the error $\mathbf{I} - \mathbf{I}^*$, when the error is null the image is completely gray).

The same experiment was now considered but with a full illumination model. As shown in Figures 18a and 18c the tracking is perfectly achieved since the error $\mathbf{I} - \mathbf{I}^*$



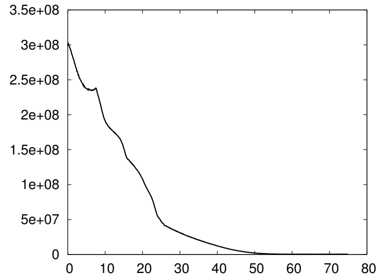
(a)

(b)

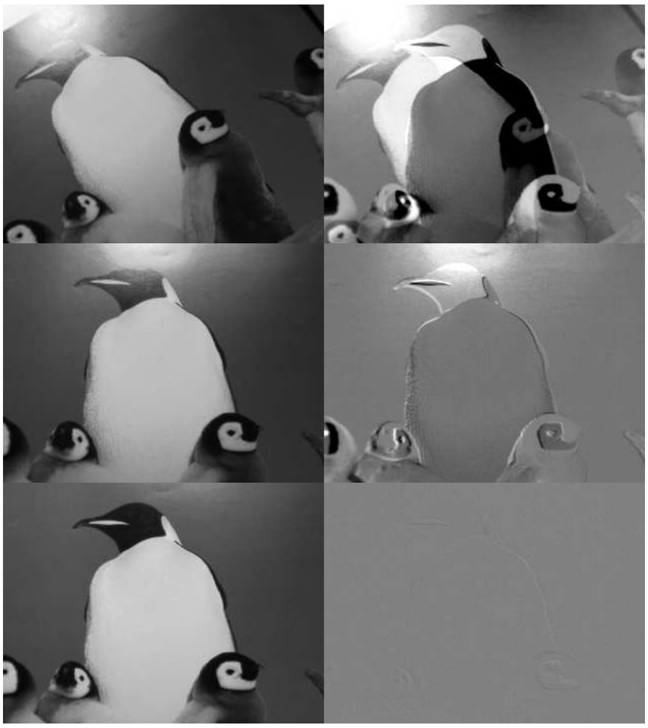
Figure 12: Positioning task with a light source motionless with respect to the object frame and located at "infinity". (a) cost function (x axis in second) (b) images acquired during the positioning task (left) and error image $\mathbf{I} - \mathbf{I}^*$ (right).

is almost null despite the occurrence of a specularity which shows the importance of such terms in the tracking process. When the velocity is constant the object is perfectly tracked as can be seen on Figure 18a where $\| \mathbf{I} - \mathbf{I}^* \|$ is depicted. Error in the image remains small except when the object stops or accelerates (see the peaks in Figure 18a). The camera velocity (see Figure 18b) shows a pure motion along the x ($\pm$ 1cm/s) axis that corresponds to the ground truth. For each pixel, except during accelerations and decelerations, $|I - I^*| < 5$.

In the second experiment, we move the object by hand as seen on the first row of Figure 19c. The related images acquired by the camera are shown on the second row. We then have a more complex 3D object motion. Note that all the 6 d.o.f. of the robot are controlled (see Fig. 19b). The error $\| \mathbf{I} - \mathbf{I}^* \|$ is shown in Figure 19a while $\mathbf{I} - \mathbf{I}^*$



(a)



(b)

Figure 13: Positioning task with a light source motionless with respect to the object frame and located at "infinity". (a) cost function (x axis in second) (b) images acquired during the positioning task (left) and error image $\mathbf{I} - \mathbf{I}^*$ (right).

is shown on the third line of Figure 19c. When the object is moving the error is more important than during the previous experiment. This is due to the fact that, since the object velocity is obviously no more constant, it is no longer possible to consider an integral term leading to classical tracking errors. In contrast, when the motion stops the camera moves to reduce these errors (iterations 5300 and 9500).

6 Conclusion and future works

We have shown in this report that it is possible to use directly the luminance of all the pixels in an image as visual features in visual servoing. To the best of our knowledge

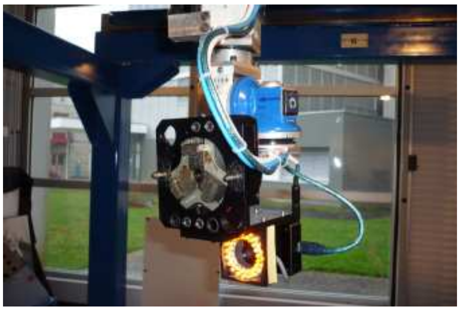


Figure 14: Camera and light-ring mounted on the robot end-effector.

this is the first time that visual servoing has been handled without any image processing (except the image spatial gradient required for the computation of the interaction matrix) nor learning step. Indeed, unlike classical visual servoing where geometrical features are used, using photometric visual servoing does not need any matching between the initial and desired features, nor between the current and the previous features. It is a very important issue when complex scenes have to be considered. To do that, the interaction matrix has been analytically computed. This computation is based on an illumination model able to tackle complex illumination variations of the scene, it is also able to tackle non Lambertian scenes. Our approach has been validated on various scenes and various lightings (diffuse or not) as well as on positioning or tracking tasks. Concerning positioning tasks, the positioning error is always very low. Supplementary advantages are that our approach is not sensitive to partial occlusions and to coarse approximations of the depths required to compute the interaction matrix. Let us point out that even in cases of non Lambertian scenes, the simple interaction matrix based on the temporal luminance constancy hypothesis leads to a good behavior and to very low positioning errors. Future work will concern the case when the intensity of the lighting source may vary during the servoing.

Appendix

Denoting $\varphi = \{x, y, z\}$, one element of (15) writes as

$$R_\varphi = 2u_2 n_\varphi - L_\varphi \tag{79}$$

which gives

$$\boldsymbol{\nabla} R_\varphi = 2\big(n_\varphi \boldsymbol{\nabla} u_2 + u_2 \boldsymbol{\nabla} n_\varphi\big) - \boldsymbol{\nabla} L_\varphi \tag{80}$$

leading to

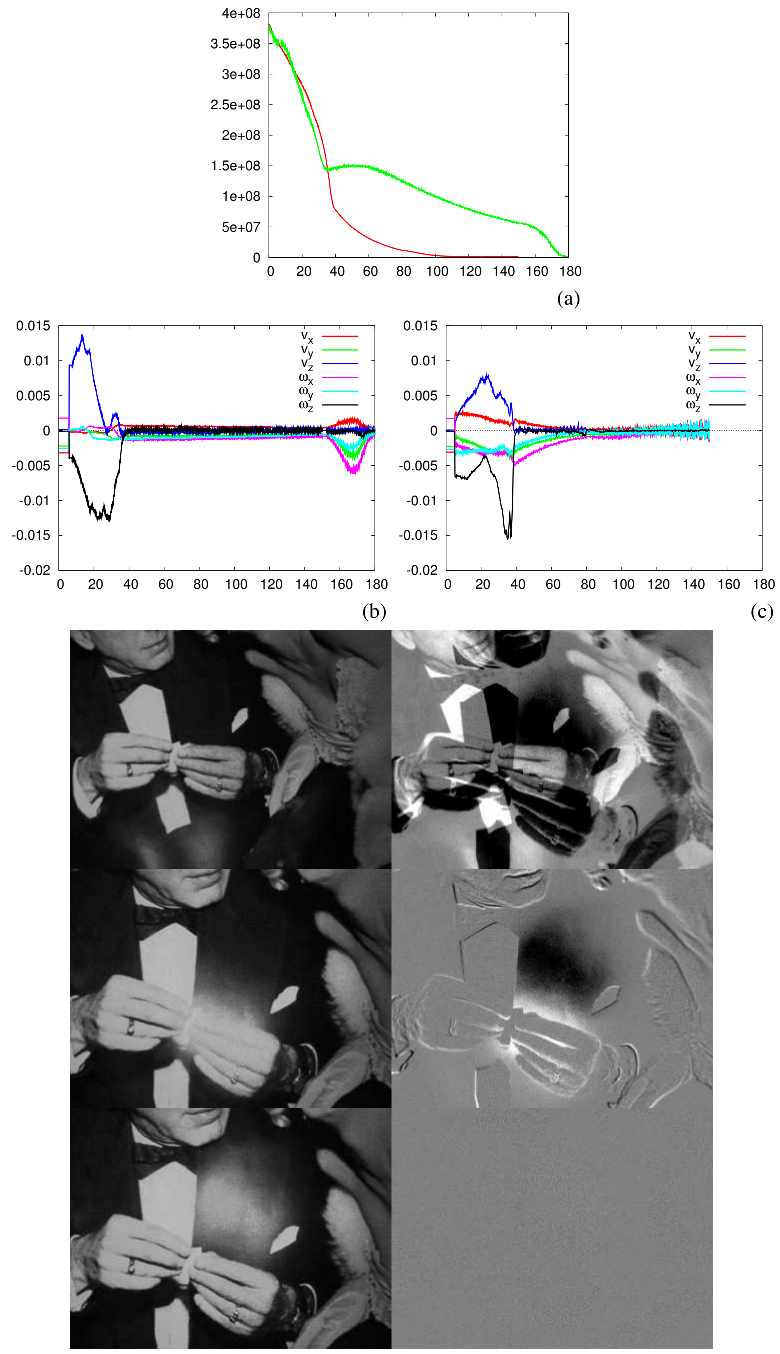
$$\mathbf{J}^{\mathbf{R}} = \begin{bmatrix} \boldsymbol{\nabla} R_x^\top \\ \boldsymbol{\nabla} R_y^\top \\ \boldsymbol{\nabla} R_z^\top \end{bmatrix} = 2\mathbf{n} \boldsymbol{\nabla} u_2^\top + 2u_2 \mathbf{J}^{\mathbf{n}} - \mathbf{J}^{\mathbf{L}} \tag{81}$$

where $\boldsymbol{\nabla} u_2$ is the spatial gradient of u_2 given by

$$\boldsymbol{\nabla} u_2 = \mathbf{J}^{\mathbf{n}\top} \mathbf{L} + \mathbf{J}^{\mathbf{L}\top} \mathbf{n}. \tag{82}$$

that finally yields

$$\mathbf{J}^{\mathbf{R}} = 2\mathbf{n}\big(\mathbf{L}^\top \mathbf{J}^{\mathbf{n}} + \mathbf{n}^\top \mathbf{J}^{\mathbf{L}}\big) + 2u_2 \mathbf{J}^{\mathbf{n}} - \mathbf{J}^{\mathbf{L}}. \tag{83}$$



(a) (b) (c) (d)

Figure 15: Positioning task with the light source mounted on the camera. (a) cost function assuming a temporal luminance constancy model (green) and using an illumination model (red); (b) Camera velocity assuming a temporal luminance constancy; (c) Camera velocity using an illumination model. (d) Images acquired during the positioning task (left) and error image $\mathbf{I} - \mathbf{I}^*$ (right).

Figure 16: View of the lighting, the camera and the object to track. The object (a photo) is attached to a motorized rail that allows to control it motion.

Figure 17: First experiment: Tracking considering the interaction matrix under temporal luminance constancy hypothesis. As can be seen the tracking task failed quickly. First row shows the image acquired by the camera while $\mathbf{I} - \mathbf{I}^*$ is shown on the second row.

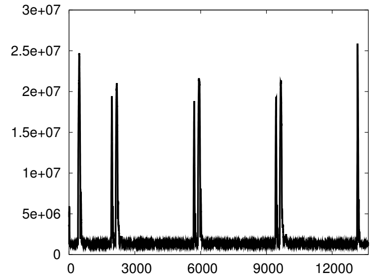
References

[1] F. Chaumette and S. Hutchinson, "Visual servoing and visual tracking," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, ch. 24, pp. 563–583.

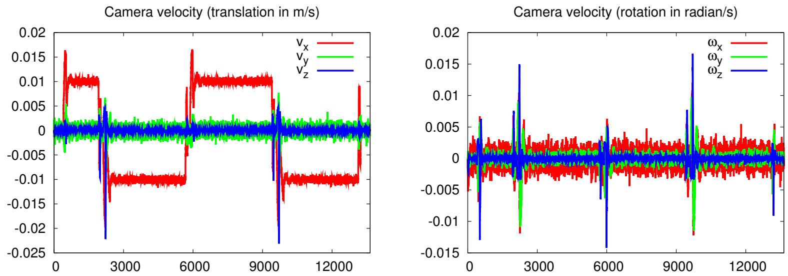
[2] E. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 53–70, June 2005, special issue on "Advances in Robot Vision", D. Kragic, H. Christensen (Eds.).

[3] J. Feddema, C. Lee, and O. Mitchell, "Automatic selection of image features for visual servoing of a robot manipulator," in *IEEE Int. Conf. on Robotics and Automation, ICRA'89*, vol. 2, Scottsdale, Arizona, May 1989, pp. 832–837.

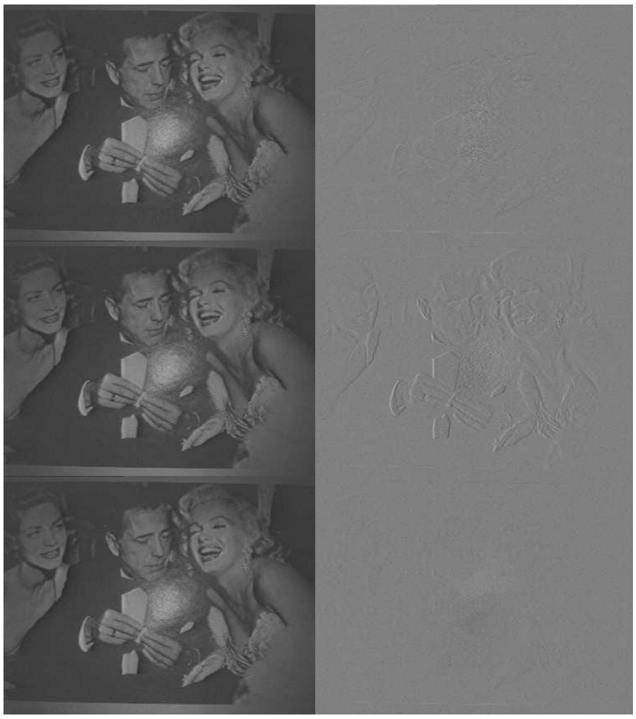
[4] F. Janabi-Sharifi and W. Wilson, "Automatic selection of image features for visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 13, no. 6, pp. 890–903, December 1997.



(a)

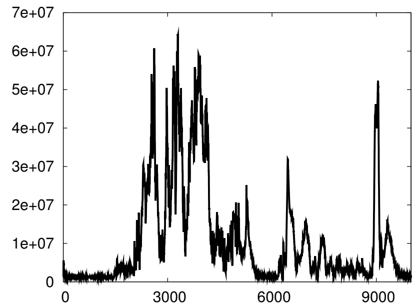


(b)

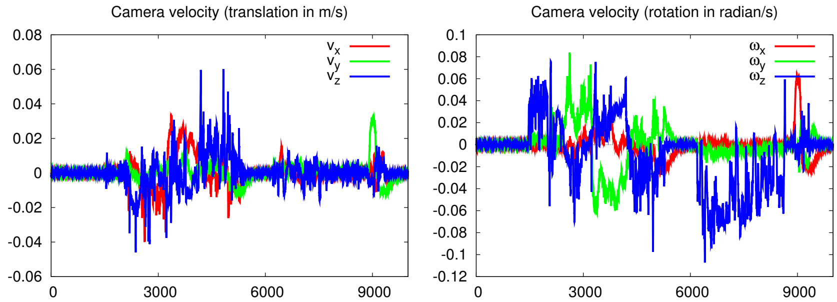


(c)

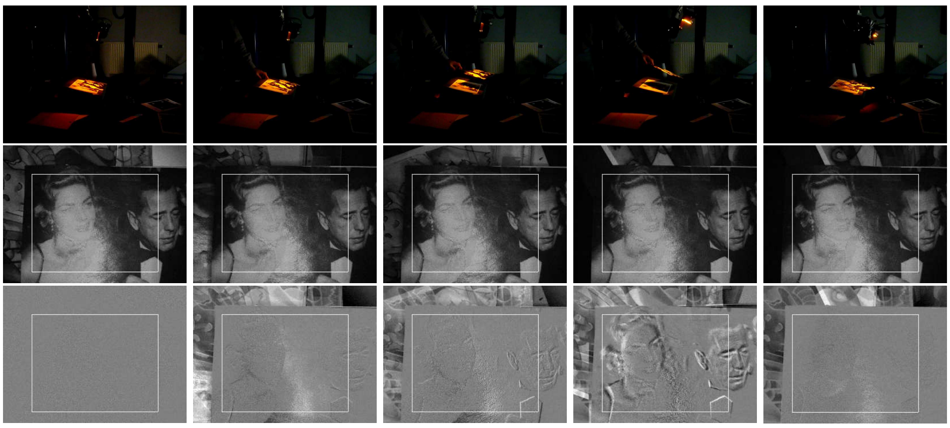
Figure 18: First experiment: tracking considering the complete interaction matrix that integrates specularity, diffuse and ambient terms (x axis in frame number). (a) Error $\| \mathbf{I} - \mathbf{I}^* \|$, (b) Camera velocity (m/s and radian/s), (c) Images at different time (left) and corresponding errors $\mathbf{I} - \mathbf{I}^*$ (right).



(a)



(b)



(c)

Figure 19: Second experiment: tracking task considering the complete interaction matrix that integrates specularity, diffuse and ambient terms (x axis in frame number). (a) Error $\| \mathbf{I} - \mathbf{I}^* \|$, (b) Camera velocity (m/s and radian/s), (c) External views of the scene at different time (first row), images at different time (second row) and corresponding errors $\mathbf{I} - \mathbf{I}^*$ (last row).

[5] N. Papanikolopoulos, "Selection of features and evaluation of visual measurements during robotic visual servoing tasks," *Journal of Intelligent and Robotic Systems*, vol. 13, pp. 279–304, October 1995.

[6] A. Comport, E. Marchand, and F. Chaumette, "Statistically robust 2D visual servoing," *IEEE Trans. on Robotics*, vol. 22, no. 2, pp. 415–421, apr 2006.

[7] P. Questa, E. Grossmann, and G. Sandini, "Camera self orientation and docking maneuver using normal flow," in *SPIE AeroSense'95*, vol. 2488, Orlando, Florida, USA, April 1995, pp. 274–283.

[8] V. Sundareswaran, P. Bouthemy, and F. Chaumette, "Exploiting image motion for active vision in a visual servoing framework," *Int. Journal of Robotics Research*, vol. 15, no. 6, pp. 629–645, June 1996.

[9] J. Santos-Victor and G. Sandini, "Visual behaviors for docking," *Computer Vision and Image Understanding*, vol. 67, no. 3, pp. 223–238, September 1997.

[10] A. Crétual and F. Chaumette, "Visual servoing based on image motion," *Int. Journal of Robotics Research*, vol. 20, no. 11, pp. 857–877, November 2001.

[11] S. Nayar, S. Nene, and H. Murase, "Subspace methods for robot vision," *IEEE Trans. on Robotics*, vol. 12, no. 5, pp. 750–758, October 1996.

[12] K. Deguchi, "A direct interpretation of dynamic images with camera and object motions for vision guided robot control," *Int. Journal of Computer Vision*, vol. 37, no. 1, pp. 7–20, June 2000.

[13] V. Kallem, M. Dewan, J. Swensen, G. Hager, and N. Cowan, "Kernel-based visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS'07*, San Diego, USA, October 2007.

[14] A. Abdul Hafez, S. Achar, and C. Jawahar, "Visual servoing based on gaussian mixture models," in *IEEE Int. Conf. on Robotics and Automation, ICRA'08*, Pasadena, California, May 2008.

[15] S. Benhimane and E. Malis, "Homography-based 2d visual tracking and servoing," *Int. Journal of Robotics Research*, vol. 26, no. 7, pp. 661–676, July 2007.

[16] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, August 1981.

[17] A. Verri and T. Poggio, "Motion field and optical flow: qualitative properties," *IEEE Trans. on PAMI*, vol. 11, no. 5, pp. 490–498, May 1989.

[18] R. Woodham, "Photometric method for determining surface orientation from multiple images," *Optical Engineering*, vol. 19, no. 1, pp. 139–144, 1980.

[19] K. Ikeuchi, "Determining surface orientations of specular surfaces by using the photometric stereo method," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 661–669, July 1981.

[20] P. Hallinan, "A low-dimensional representation of human faces for arbitrary lighting conditions," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Seattle, Washington, June 1994, pp. 995–999.

[21] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, October 1998.

[22] M. La Cascia, S. Sclaroff, and V. Athitsos, “Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, April 2000.

[23] M. Black, D. Fleet, and Y. Yacoob, “Robustly estimating changes in image appearance,” *Computer Vision and Image Understanding*, vol. 78, pp. 8–31, 2000.

[24] G. Silveira and E. Malis, “Real-time visual tracking under arbitrary illumination changes,” in *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR’07*, Minneapolis, USA, June 2007, pp. 1–6.

[25] S. Negahdaripour, “Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 9, pp. 961–979, 1998.

[26] H. Haussecker and D. Fleet, “Computing optical flow with physical models of brightness variation,” *IEEE Trans. on PAMI*, vol. 23, no. 6, pp. 661–673, June 2000.

[27] J. Reichmann, “Determination of absorption and scattering coefficients for non homogeneous media,” *Applied Optics*, vol. 12, pp. 1811–1815, 1973.

[28] P. Beckmann and A. Spizzichino, *The scattering of electromagnetic waves from rough surfaces*, 2nd ed. Artech House Inc, 1987.

[29] K. Torrance and E. Sparrow, “Theory for off-specular reflection from roughened surfaces,” *Journal of the Optical Society of America*, vol. 57, pp. 1105–1114, 1967.

[30] B. Phong, “Illumination for computer generated pictures,” *Communication of the ACM*, vol. 18, no. 6, pp. 311–317, June 1975.

[31] J. Blinn, “Models of light reflection for computer synthesized pictures,” in *ACM Conf. on Computer graphics and interactive techniques, SIGGRAPH’77*, San Jose, California, 1977, pp. 192–198.

[32] C. Collewet and E. Marchand, “Modeling complex luminance variations for target tracking,” in *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR’08*, Anchorage, Alaska, June 2008.

[33] E. Malis, “Improving vision-based control using efficient second-order minimization techniques,” in *IEEE Int. Conf. on Robotics and Automation, ICRA’04*, vol. 2, New Orleans, April 2004, pp. 1843–1848.

[34] K. Hashimoto and H. Kimura, “Lq optimal and non-linear approaches to visual servoing,” in *Visual Servoing*, K. Hashimoto, Ed. Singapour: World Scientific Series in Robotics and Automated Systems, 1993, vol. 7, pp. 165–198.

[35] J.-T. Lapresté and Y. Mezouar, “A Hessian approach to visual servoing,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS’04*, vol. 1, Sendai, Japan, October 2004, pp. 998–1003.

[36] O. Tahri and Y. Mezouar, “On the efficient second order minimization and image-based visual servoing,” in *IEEE Int. Conf. on Robotics and Automation*, Pasadena, California, May 2008, pp. 3213–3218.

[37] F. Chaumette, P. Rives, and B. Espiau, “Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing,” in *IEEE Int. Conf. on Robotics and Automation*, vol. 3, Sacramento, California, USA, April 1991, pp. 2248–2253.

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399